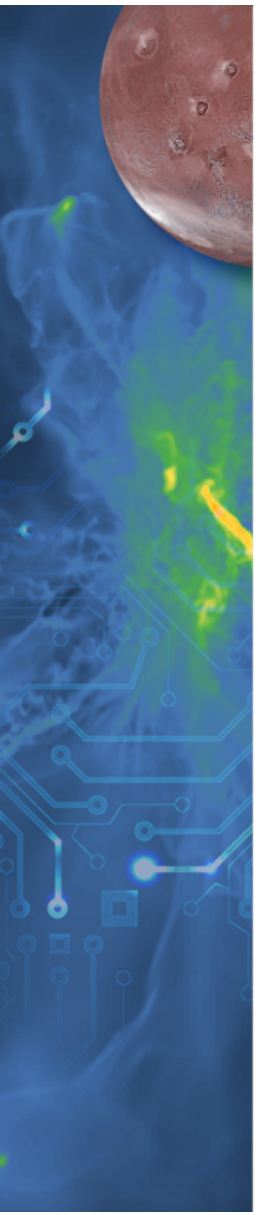
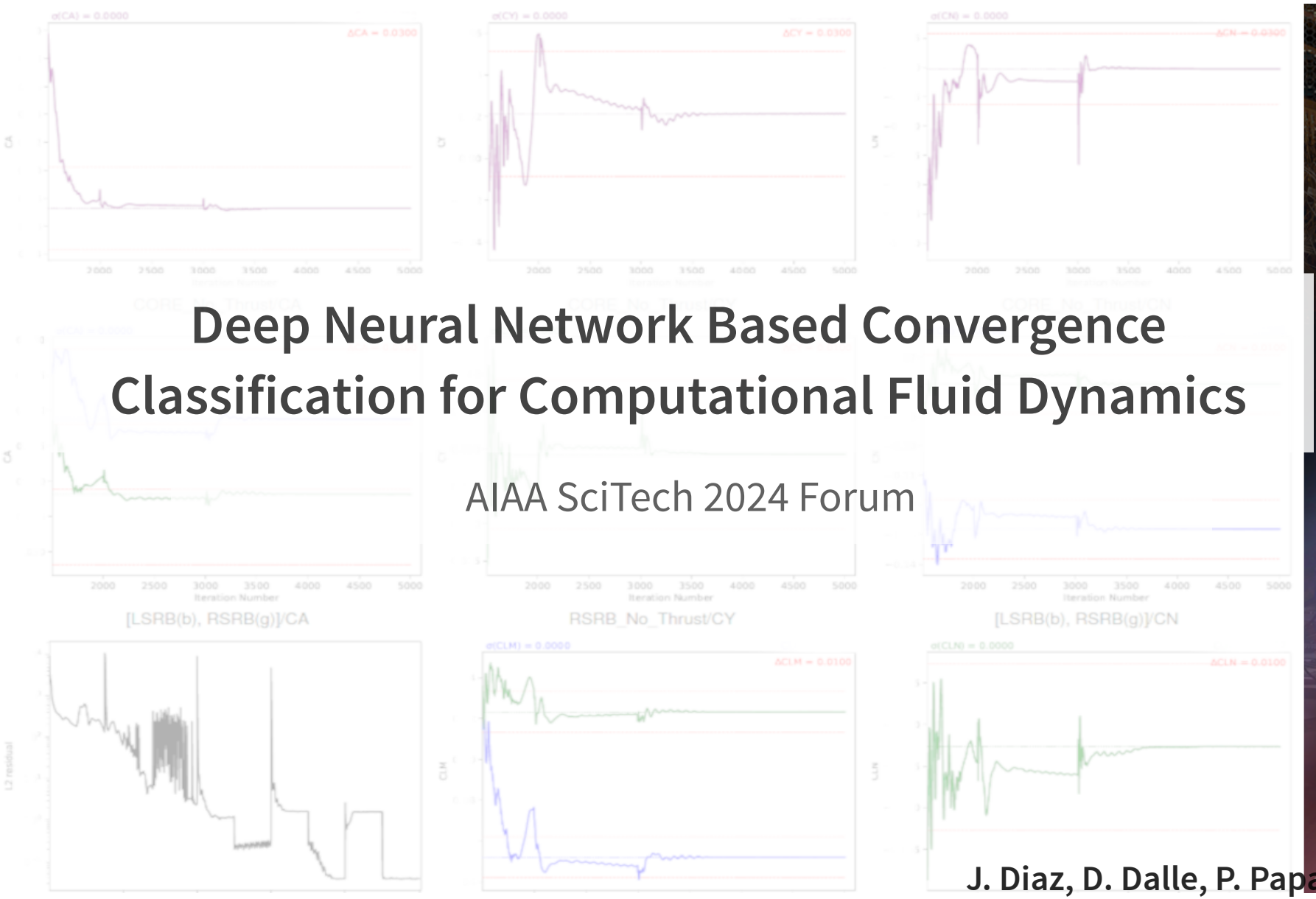
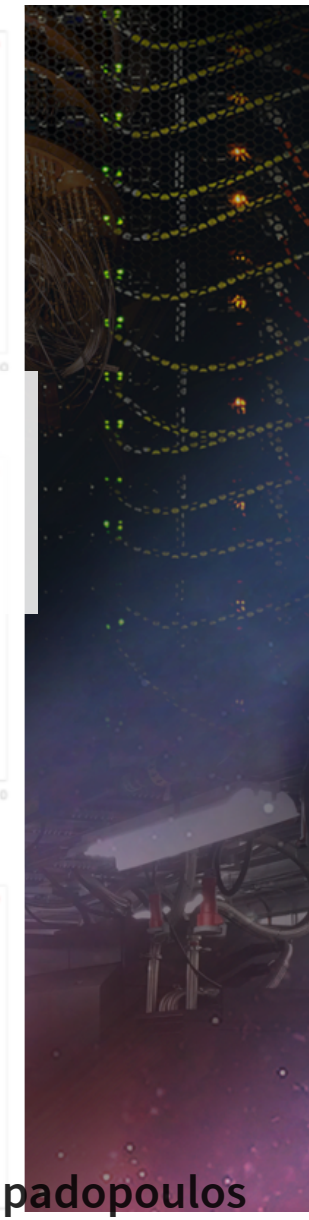






Deep Neural Network Based Convergence Classification for Computational Fluid Dynamics

AIAA SciTech 2024 Forum

J. Diaz, D. Dalle, P. Papadopoulos

January 12, 2024

Presentation Outline

1. Motivation
2. Introduction to PYCNIC
3. Look under the hood
4. Results
5. Future work

What I do…

Ascent and **Booster Separation** modeling of NASA's Space Launch System (SLS) vehicle.

These are **LARGE** aerodynamic databases using: OVERFLOW, CART3D, and FUN3D.

The solver FUN3D is the focus of this presentation.

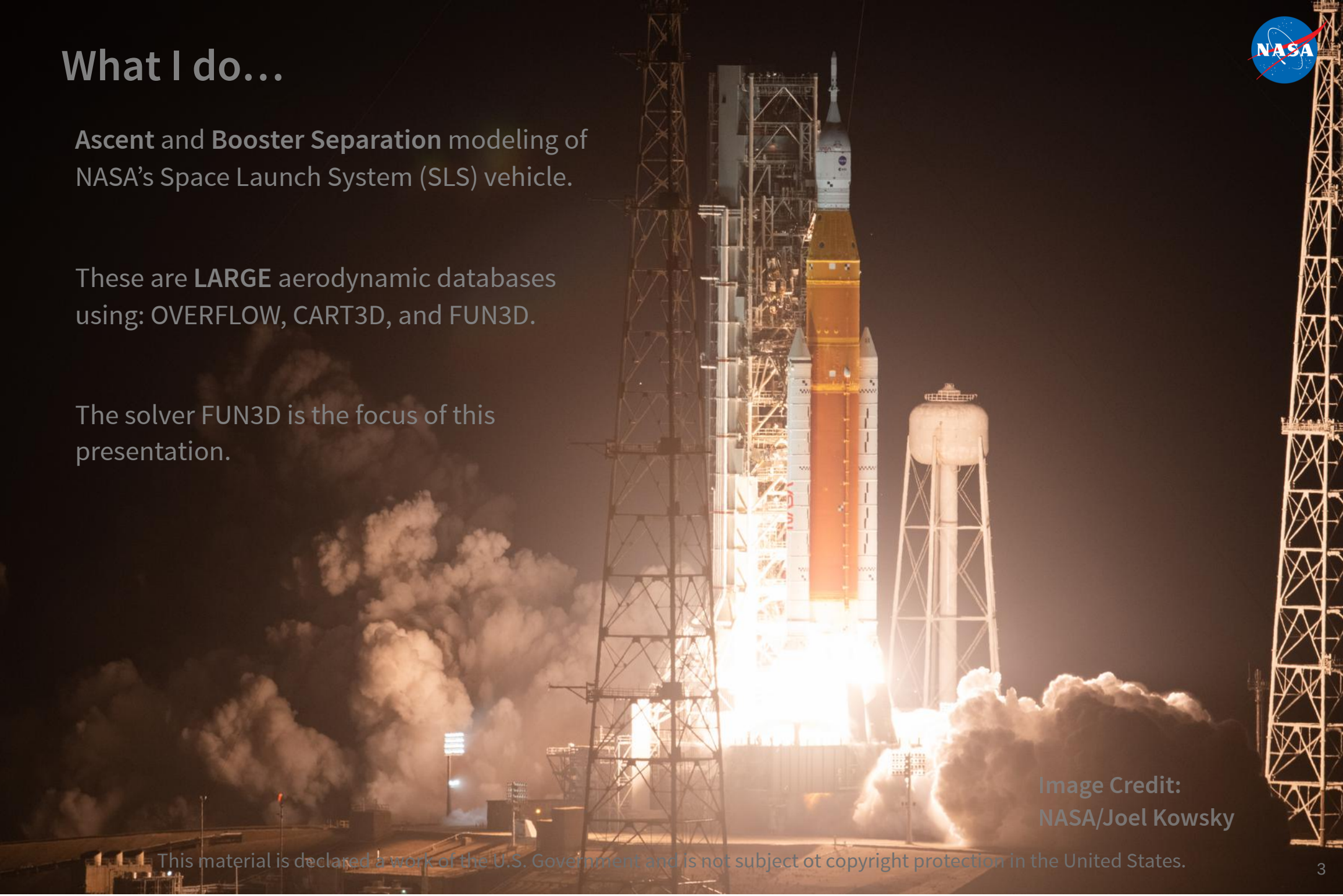


Image Credit:
NASA/Joel Kowsky

Current Workflow

1) Check cases on Pleiades Front End

```
pfe27:f3_vac1$ pyfun -f run/poweron02-ML.json --cons "user=='@jfdiaz3'" -I 0:10 -c
```

Current Workflow

1) Check cases on Pleiades Front End

```
pfe27:f3_vac1$ pyfun -f run/poweron02-ML.json --cons "user=='@jfdiaz3'" -I 0:10 -c

Importing module 'f3drunasc'
Importing module 'f3dnas'
Importing module 'tnas45pal'
  InitFunction: f3dnas.init_config()
  InitFunction: tnas45pal.add_triqfm()
Case Config/Run Directory           Status  Iterations  Que CPU Time
---- ------------------------------ ------- ----------- --- --------
0    poweron02-ML/m0.50a0.0r000.0_a DONE    12500/12500 .   145224.7
1    poweron02-ML/m0.50a1.0r000.0   RUN     4233/4500   .    29998.9
2    poweron02-ML/m0.50a2.0r000.0   DONE    7000/7000   .    33468.2
3    poweron02-ML/m0.50a6.0r000.0   DONE    5500/5500   .    37828.9
4    poweron02-ML/m0.70a0.0r000.0   DONE    10500/10500 .    75882.0
5    poweron02-ML/m0.70a1.0r000.0   DONE    6500/6500   .    46045.2
6    poweron02-ML/m0.70a2.0r000.0   DONE    4750/4750   .    31409.3
7    poweron02-ML/m0.70a6.0r000.0   ERROR   3500/4500   .    19231.8
8    poweron02-ML/m0.80a0.0r000.0   DONE    8500/8500   .    60833.1
9    poweron02-ML/m0.80a1.0r000.0   DONE    5500/5500   .    38438.4
```

Current Workflow

1) Check cases on Pleiades Front End

```
pfe27:f3_vac1$ pyfun -f run/poweron02-ML.json --cons "user=='@jfdiaz3'" -I 0:10 -c
```

Current Workflow

1) Check cases on Pleiades Front End

```
pfe27:f3_vac1$ pyfun -f run/poweron02-ML.json --cons "user=='@jfdiaz3'" -I 0:10 -c
```

2) Generate case report

```
1  pfe27:f3_vac1$ pyfun -f run/poweron02-ML.json --cons "user=='@jfdiaz3'" --report -I 0,2:7,8:10
```

Current Workflow

1) Check cases on Pleiades Front End

```
pfe27:f3_vac1$ pyfun -f run/poweron02-ML.json --cons "user=='@jfdiaz3'" -I 0:10 -c
```

2) Generate case report

```
pfe27:f3_vac1$ pyfun -f run/poweron02-ML.json --cons "user=='@jfdiaz3'" --report -I 0,2:7,8:10

Importing module 'f3drunasc'
Importing module 'f3dnas'
Importing module 'tnas45pal'
  InitFunction: f3dnas.init_config()
  InitFunction: tnas45pal.add_triqfm()
poweron02-ML/m0.50a0.0r000.0_a
  SRB_AftBSMPod_CA: New subfig at iteration 12500
QStandardPaths: XDG_RUNTIME_DIR not set, defaulting to '/tmp/runtime-jfdiaz3'
  SRB_AftBSMPod_CN: New subfig at iteration 12500
  SRB_MainTun_CA: New subfig at iteration 12500
  STACK-phi090-mach-large: New subfig at iteration 12500
> tec360 -b -p export-lay.mcr -mesa
    (PWD = 'poweron02-ML/m0.50a0.0r000.0_a/')
    (STDOUT = 'tecmcr.out')
  STACK-phi000-mach-large: New subfig at iteration 12500
> tec360 -b -p export-lay.mcr -mesa
    (PWD = 'poweron02-ML/m0.50a0.0r000.0_a/')
    (STDOUT = 'tecmcr.out')
  STACK_Aft-phi090-mach-large: New subfig at iteration 12500
> tec360 -b -p export-lay.mcr -mesa
    (PWD = 'poweron02-ML/m0.50a0.0r000.0_a/')
    (STDOUT = 'tecmcr.out')

    ...
```

Current Workflow

1) Check cases on Pleiades Front End

```
pfe27:f3_vac1$ pyfun -f run/poweron02-ML.json --cons "user=='@jfdiaz3'" -I 0:10 -c
```

2) Generate case report

```
1  pfe27:f3_vac1$ pyfun -f run/poweron02-ML.json --cons "user=='@jfdiaz3'" --report -I 0,2:7,8:10
```

Current Workflow

1) Check cases on Pleiades Front End

```
pfe27:f3_vac1$ pyfun -f run/poweron02-ML.json --cons "user=='@jfdiaz3'" -I 0:10 -c
```

2) Generate case report

```
pfe27:f3_vac1$ pyfun -f run/poweron02-ML.json --cons "user=='@jfdiaz3'" --report -I 0,2:7,8:10
```

3) Transfer report to local machine

```
pfe27:f3_vac1$ putsup report/report-f3d-poweron02-MLpdf
nasmac3329:28008$ getsup
```

Current Workflow

1) Check cases on Pleiades Front End

```
pfe27:f3_vac1$ pyfun -f run/poweron02-ML.json --cons "user=='@jfdiaz3'" -I 0:10 -c
```

2) Generate case report

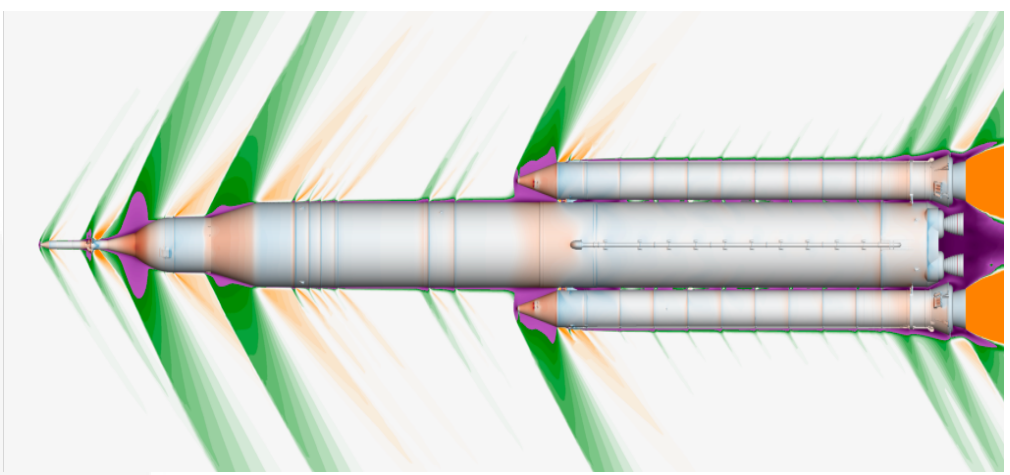
```
pfe27:f3_vac1$ pyfun -f run/poweron02-ML.json --cons "user=='@jfdiaz3'" --report -I 0,2:7,8:10
```

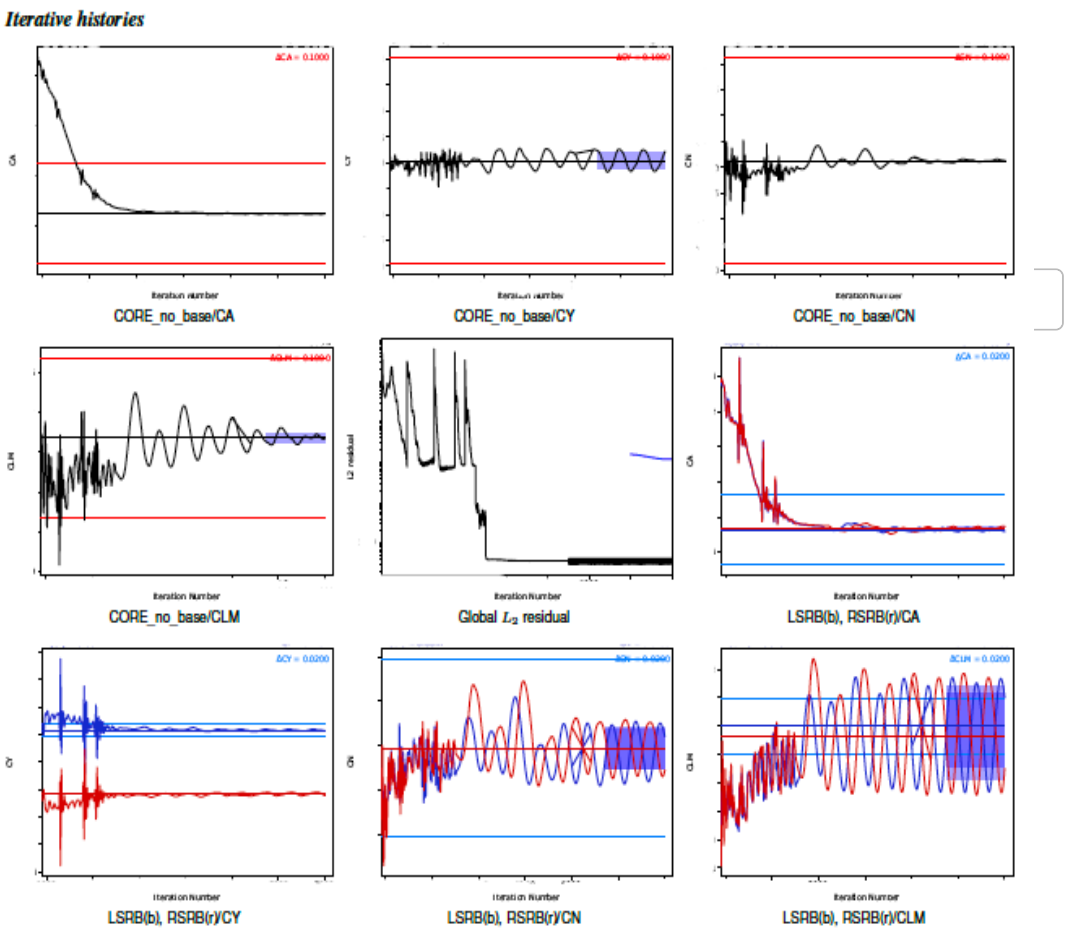
3) Transfer report to local machine

```
pfe27:f3_vac1$ putsup report/report-f3d-poweron02-MLpdf
nasmac3329:28008$ getsup
```

4) Assess each case for convergence

```
1  nasmac3329:28008$ open report/report-f3d-poweron02-ML.pdf
```





Current Workflow

1) Check cases on Pleiades Front End

```
pfe27:f3_vac1$ pyfun -f run/poweron02-ML.json --cons "user=='@jfdiaz3'" -I 0:10 -c
```

2) Generate case report

```
pfe27:f3_vac1$ pyfun -f run/poweron02-ML.json --cons "user=='@jfdiaz3'" --report -I 0,2:7,8:10
```

3) Transfer report to local machine

```
pfe27:f3_vac1$ putsup report/report-f3d-poweron02-MLpdf
nasmac3329:28008$ getsup
```

4) Assess each case for convergence

```
nasmac3329:28008$ open report/report-f3d-poweron02-ML.pdf
```

5) PASS or EXTEND cases on Pleiades

```
1  pfe27:f3_vac1$ pyfun -f run/poweron02-ML.json --cons "user=='@jfdiaz3'" -I 0,2:7 --PASS
2  pfe27:f3_vac1$ pyfun -f run/poweron02-ML.json --cons "user=='@jfdiaz3'" -I 8:11 --extend
```

SLS Block1B Ascent aerodynamic database is **1242** CFD cases.

There were an average of **4.5 manual checks** for convergence per case.

For 100 CFD cases: **HOURS**

Workflow with PYCNIC

1) Check cases on Pleiades Front End

```
pfe27:f3_vac1$ pyfun -f run/poweron02-ML.json --cons "user=='@jfdiaz3'" -I 0:10 -c
```

~~2) Generate case report~~

```
~~pfe27:f3_vac1$ pyfun -f run/poweron02-ML.json --cons "user=='@jfdiaz3'" --report -I 0,2:7,8:10~~
```

~~3) Transfer report to local machine~~

```
~~pfe27:f3_vac1$ putsup report/report-f3d-poweron02-MLpdf~~
~~nasmac3329:28008$ getsup~~
```

4) Assess each case for convergence

```
nasmac3329:28008$ open report/report-f3d-poweron02-ML.pdf
```

5) PASS or EXTEND cases on Pleiades

```
1  pfe27:f3_vac1$ pyfun -f run/poweron02-ML.json --cons "user=='@jfdiaz3'" -I 0,2:7 --PASS
2  pfe27:f3_vac1$ pyfun -f run/poweron02-ML.json --cons "user=='@jfdiaz3'" -I 8:11 --extend
```

SLS Block1B Ascent aerodynamic database is **1242** CFD cases.

There were an average of **4.5 manual checks** for convergence per case.

For 100 CFD cases: ~~**HOURS**~~ **MINUTES**

Workflow with PYCNIC

1) Check cases on Pleiades Front End

```
pfe27:f3_vac1$ pyfun -f run/poweron02-ML.json --cons "user=='@jfdiaz3'" -I 0:10 -c
```

2) Assess each case for convergence on Pleiades

```
pfe27:f3_vac1$ python3 tools/pycnic.py -f run/poweron02-ML.json -I 0,2 --comp CORE_no_base --asym 0.005
Importing module 'f3drunasc'
Importing module 'f3dnas'
Importing module 'tnas45pal'
  InitFunction: f3dnas.init_config()
  InitFunction: tnas45pal.add_triqfm()
Component - CORE_no_base
Prepping CA Iterative History 1/4
Writing cases...: 100%|████████████████████| 2/2 [00:00<00:00,  1.64it/s]
Prepping CY Iterative History 2/4
Writing cases...: 100%|████████████████████| 2/2 [00:00<00:00,  2.56it/s]
Prepping CN Iterative History 3/4
Writing cases...: 100%|████████████████████| 2/2 [00:00<00:00,  2.76it/s]
Prepping CLM Iterative History 4/4
Writing cases...: 100%|████████████████████| 2/2 [00:00<00:00,  2.79it/s]

 Loading Machine Learning Models...

Case 0: Sampling Iterations 10500-12500
CORE_no_base - Time Accurate Binary ML: EXTEND

Case 2: Sampling Iterations 5000-7000
CORE_no_base - Time Accurate Binary ML: PASS
```

3) PASS or EXTEND cases on Pleiades

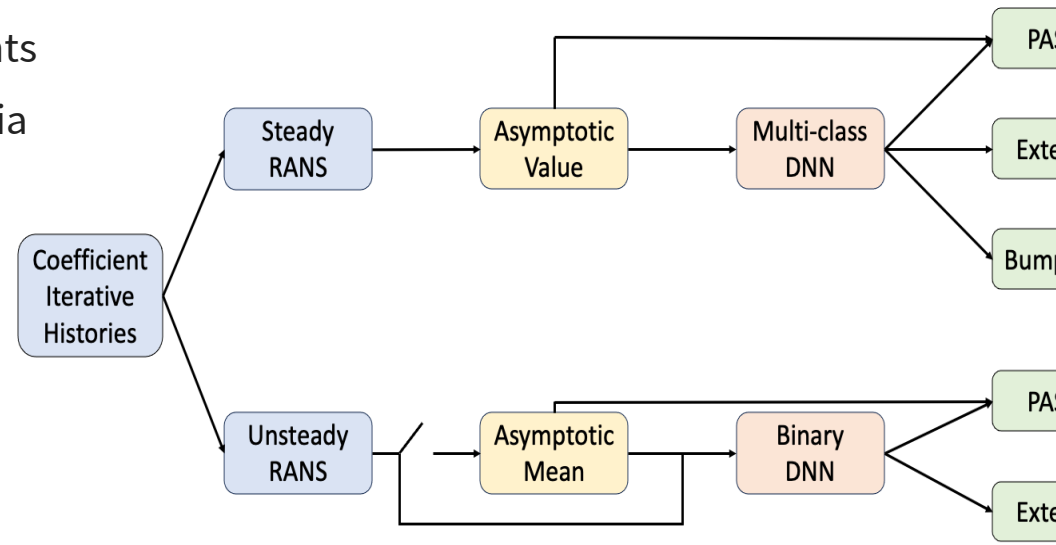
```
pfe27:f3_vac1$ pyfun -f run/poweron02-ML.json --cons "user=='@jfdiaz3'" -I 0 --extend
pfe27:f3_vac1$ pyfun -f run/poweron02-ML.json --cons "user=='@jfdiaz3'" -I 2 --PASS
```

What is PYCNIC

`PYCNIC` stands for `Python Classifier for Numerical Iterative Convergence`

and acts as an automated convergence criteria.

- Implements both **heuristic convergence criteria** as well as **supervised deep learning**
- Predictions based on multiple force and moment coefficients
- More capable than standalone heuristic convergence criteria



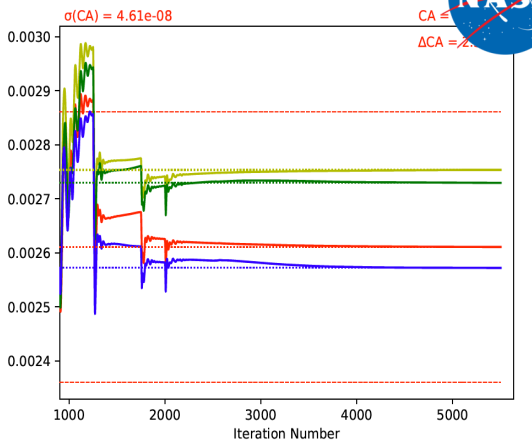
Why do we need PYCNIC

1. Expedite
    - Reduces the user's workload
    - Quickens the completion of high-volume run matrices
2. Standardize
    - Humans can have poor repeatablility when classifying the same data
    - Multiple practioners contribute with varying experience and opinions

Asymptotic Filters

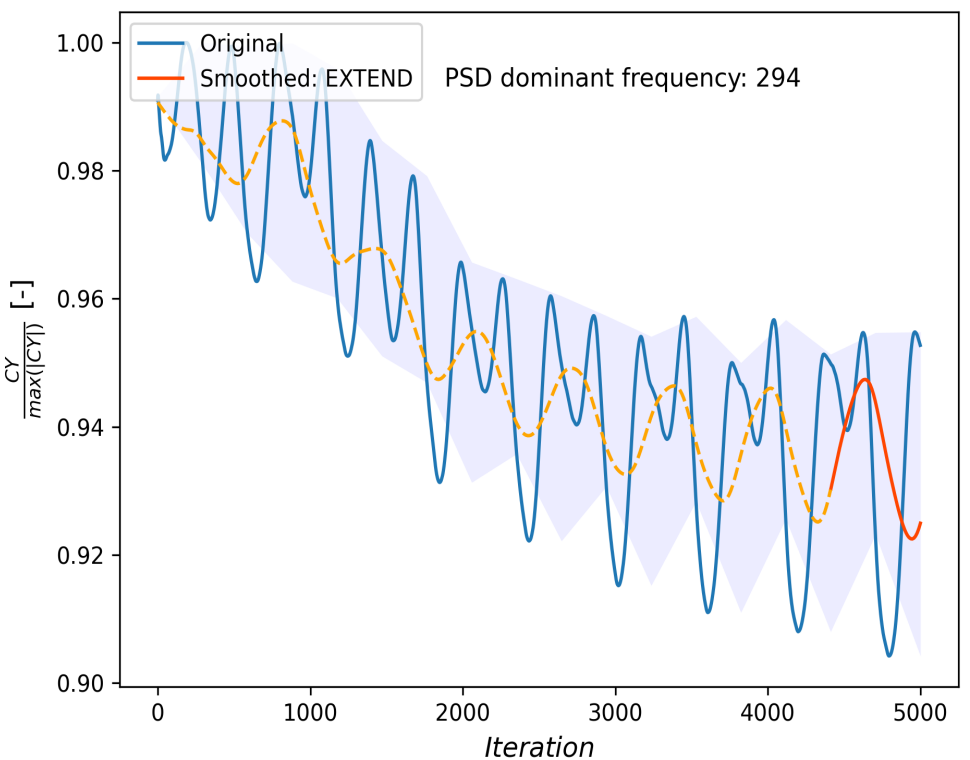
Steady RANS

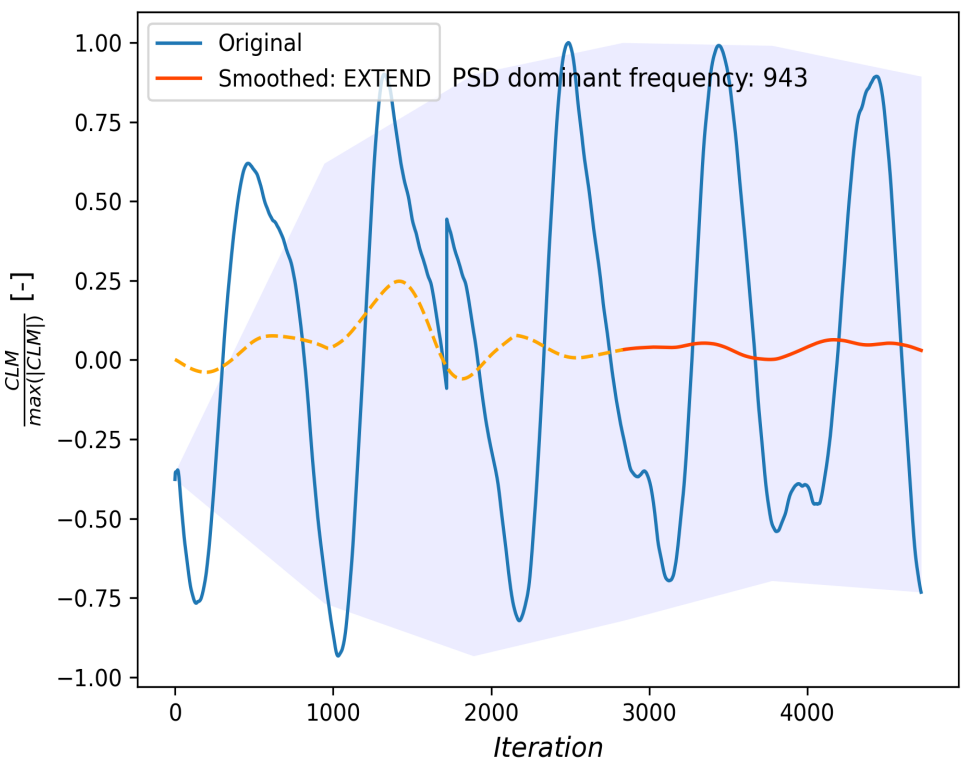
$$\begin{cases} PASS, & \text{if } |\max(C) - \min(C)| \leq X \\ \rightarrow DNN, & \text{if } |\max(C) - \min(C)| > X \end{cases}$$



Unsteady RANS

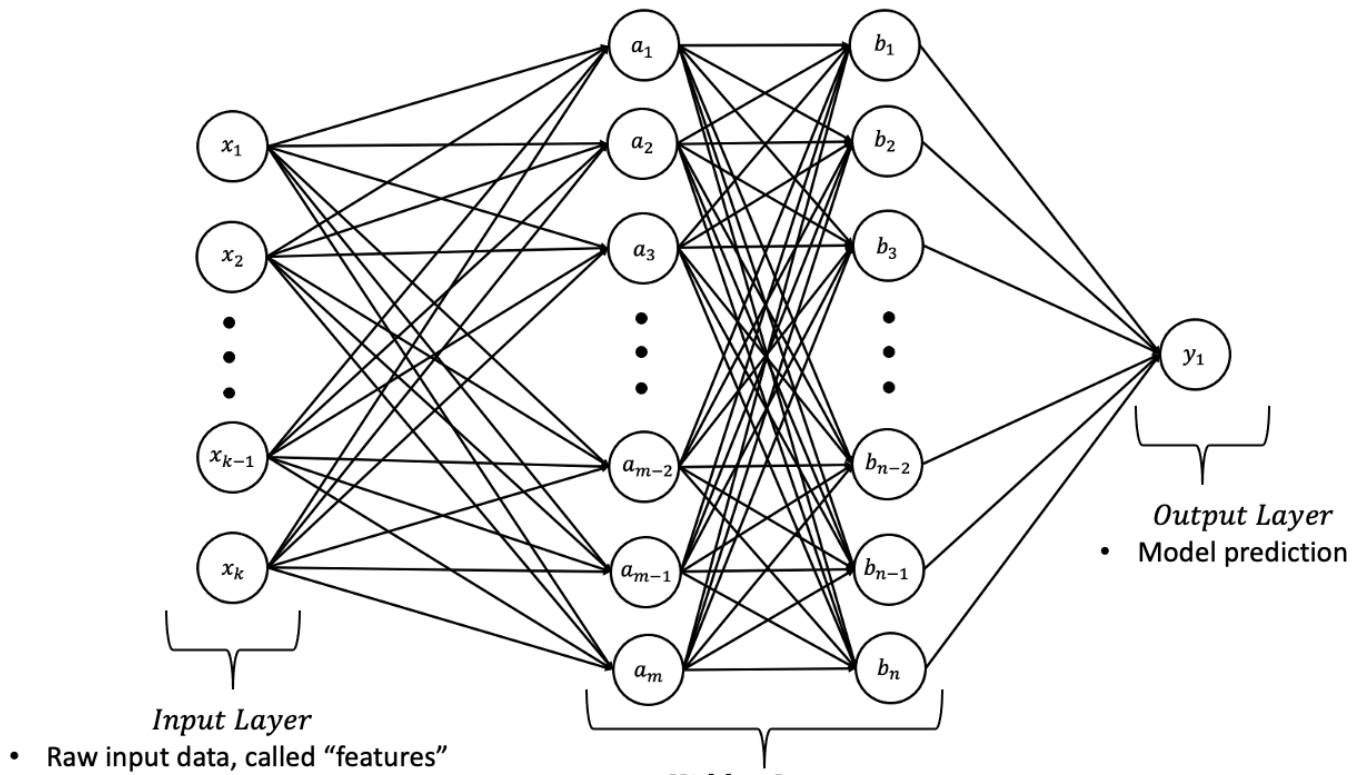
Smoothing function walks through an averaging window based on the dominant frequency determined by **Power Spectral Density** (scipy.signal.periodogram).



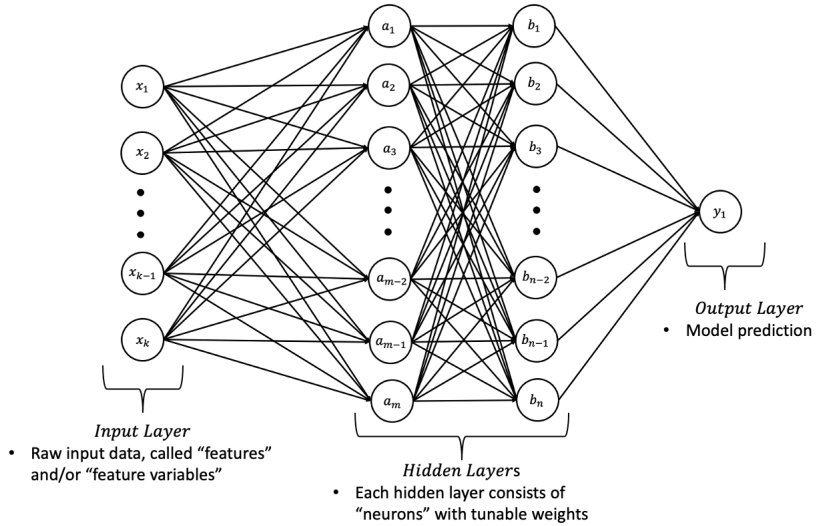


Deep Neural Network

- A Deep Neural Network (DNN) is a mathematical model inspired by the human brain which uses multiple layers to obtain a high-dimensional representation of raw input data to make a prediction.



Deep Neural Network



Deep Neural Network

Activation Functions

- **Rectified Linear Unit (ReLu)**

$$\phi(\mathbf{z}) = \begin{cases} 0, & \text{if } z_i < 0 \\ z_i, & \text{if } z_i \geq 0 \end{cases}$$

Used for all **hidden layers**.

- **Sigmoid**

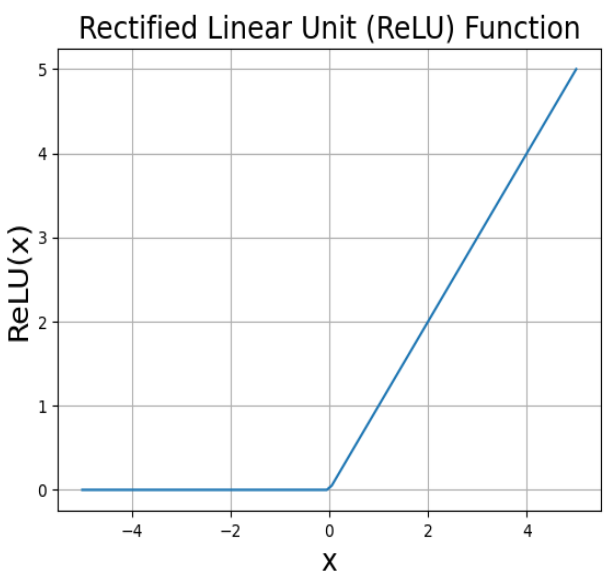
$$\sigma(\mathbf{z}) = \frac{1}{1 + e^{-\mathbf{z}}}$$

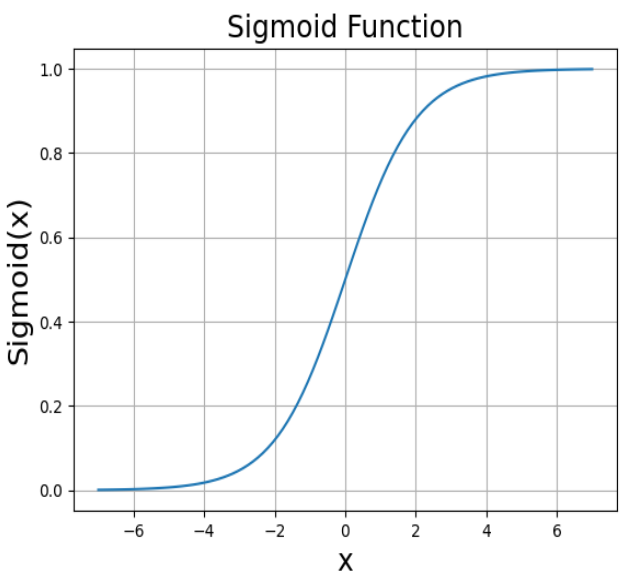
Used for **binary** model output layer. Returns a value between 0 and 1.

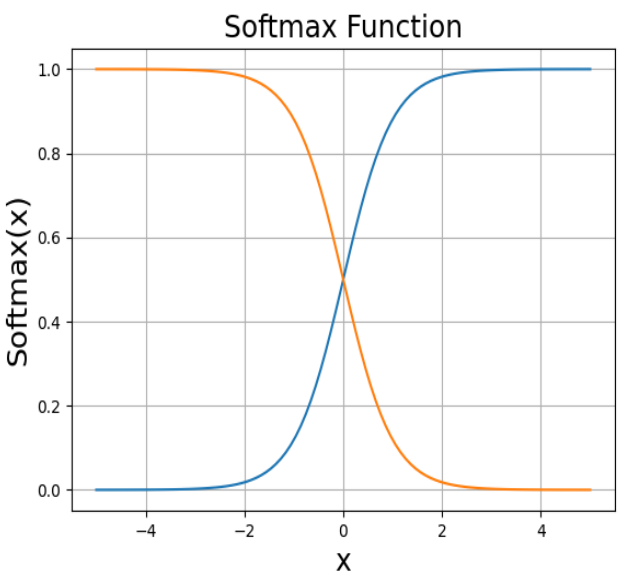
- **Softmax**

$$\sigma(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{n} e^{z_j}},$$

Used for **multi-class** model output layer. Returns a vector of values whose sum is equal to 1.







Deep Neural Network

Supervised learning approach where the training data consists of features and labels.

Aero databases perfectly lend themselves to supervised learning because they are:

1. high volume
2. cover multiple flight regimes
3. labeled

Weights are optimized through backpropagation and a stochastic gradient descent method, minimizing a loss function over multiple training epochs.

Binary Cross-entropy Loss Function

$$L_{Binary} = -\frac{1}{N}\sum_{i=1}^{N}\left(y_i \log(p_i) + (1-y_i)\log(1-p\right.$$

where, $p_i = \frac{1}{1+e^{-\mathbf{z}}}$

Sparse Catagorical Cross-entropy Loss Function

$$L_{Multi-class} = -\sum_{i=1}^{N}\log(p_i)$$

where, $p_i = \frac{e^{z_i}}{\sum_{j=1}^{n} e^{z_j}}$

- N is the number of samples.
- y_i is the true label for the i-th sample (either 0 or 1).
- p_i is the predicted probability of the i-th sample belonging to the positive class.

Deep Neural Network

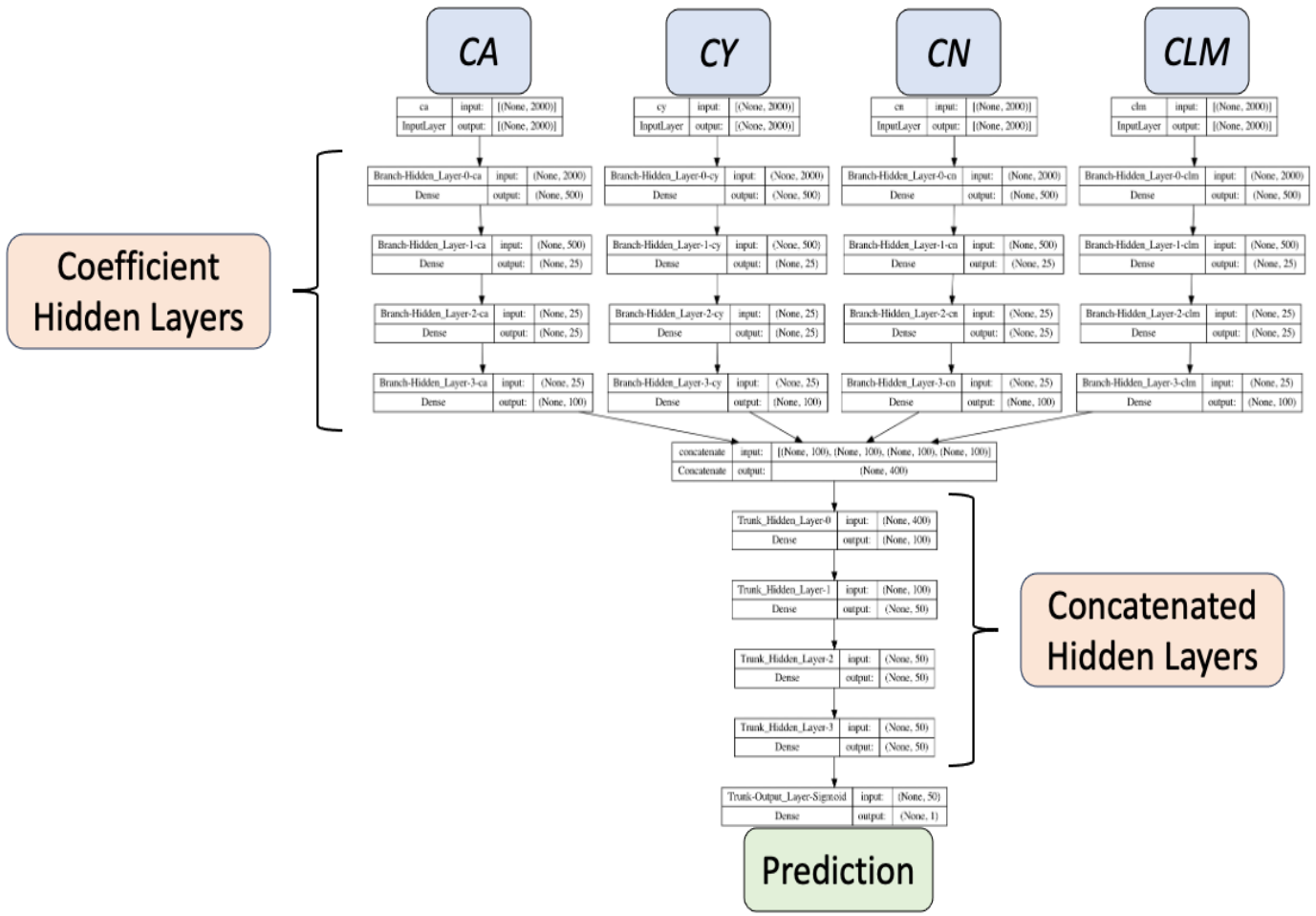
DNN Model were coded in Python using Google's machine learning framework **TensorFlow**, and optimized using the open-source library **Keras**

Input data

- 2000 features per feature variable
- Feature Variables: CA, CY, CN, CLM

Training data

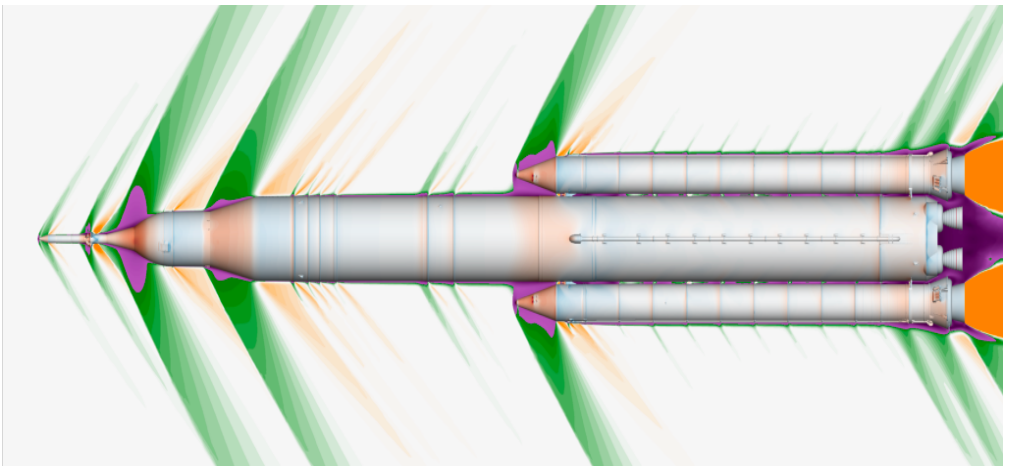
- ~300 cases per label
- 3 classes: PASS, EXTEND, BUMPTA
- optimized hyperparameters
- ~1e6 parameters
- ~36 gpu hours to train



Results: User vs Model

Model independently* ran 72 ascent cases for SLS Block1B

- Mach = [0.5 : 4.50]
- alpha = [0.0, 1.0, 2.0, 6.0]
- beta = 0.0
- **PASS** based on:
    1. `CORE_no_base`
    2. `LSRB_no_base`
    3. `RSRB_no_base`

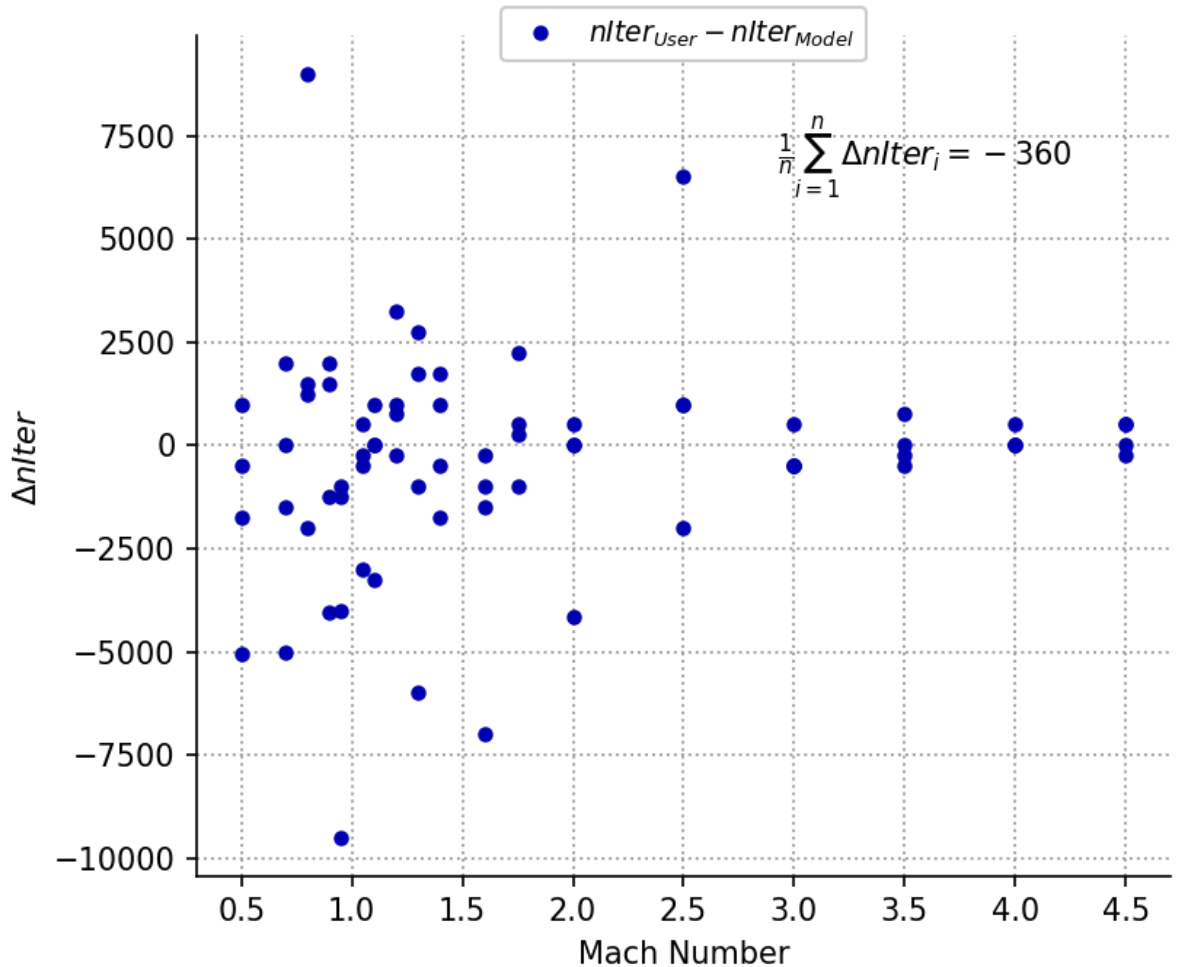


PASS Method	CORE_no_ base	RSRB_no_base	LSRB_no_base	[%]
Unsteady Asym	42	42	45	60%
Unsteady DNN	7	6	4	8%
Steady Asym	20	22	21	29%
Steady DNN	3	2	2	3%

Results: User vs Model

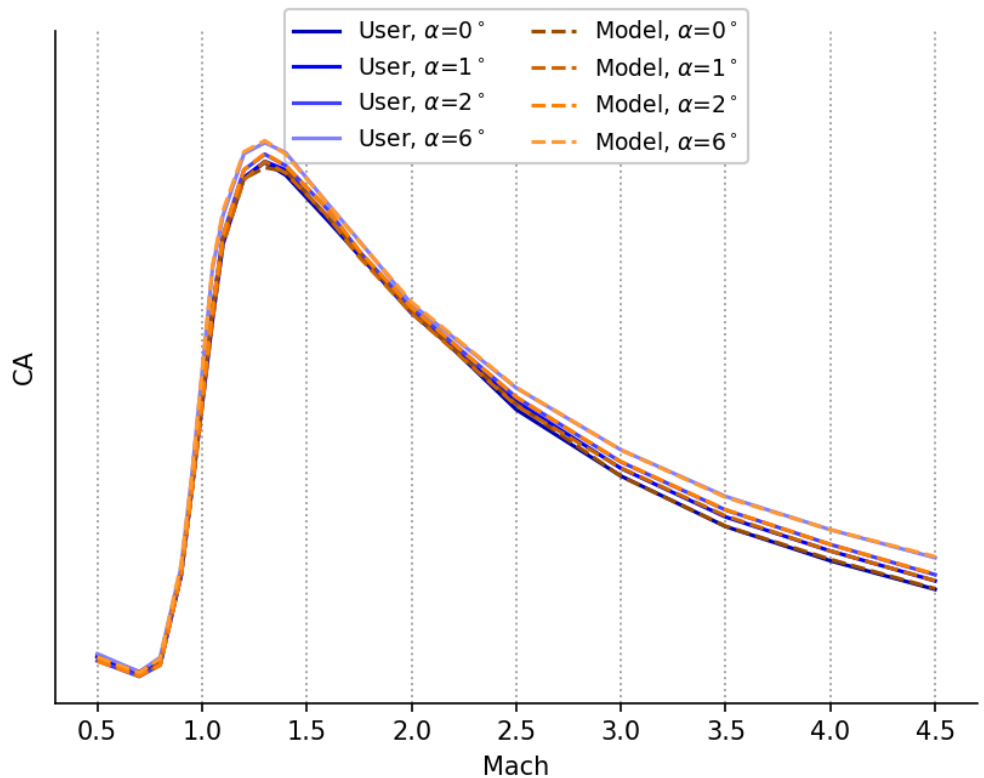
Model independently* ran 72 ascent cases for SLS Block1B

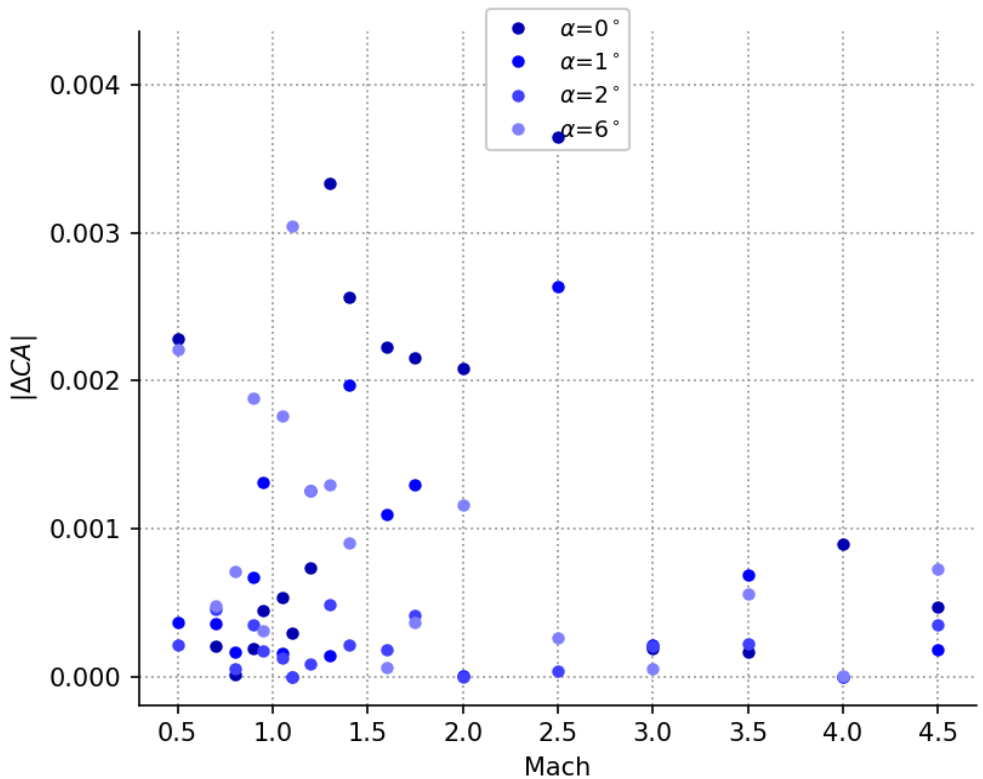
- Mach = [0.5 : 4.50]
- alpha = [0.0, 1.0, 2.0, 6.0]
- beta = 0.0
- **PASS** based on:
    1. `CORE_no_base`
    2. `LSRB_no_base`
    3. `RSRB_no_base`



Results: User vs Model

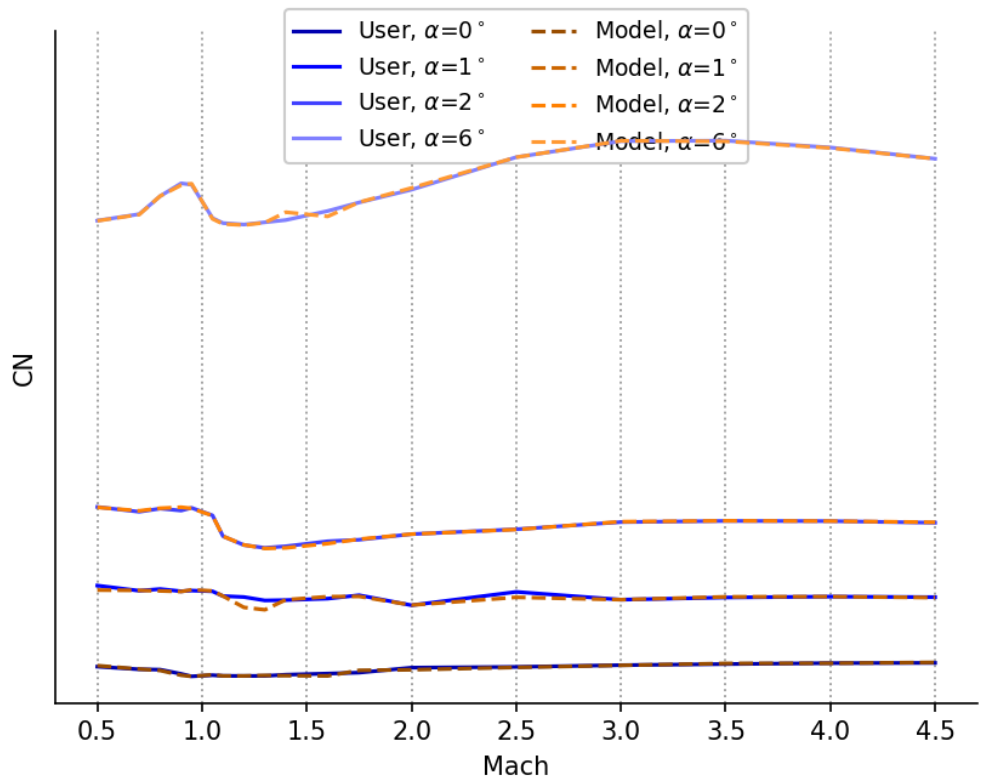
CORE

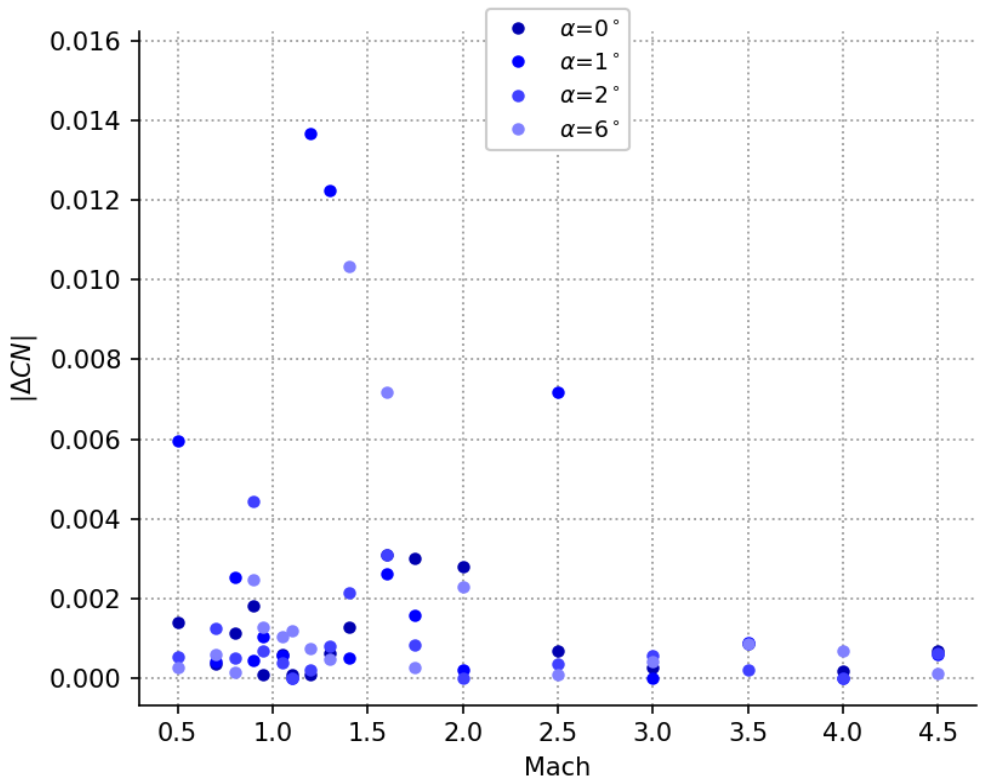




Results: User vs Model

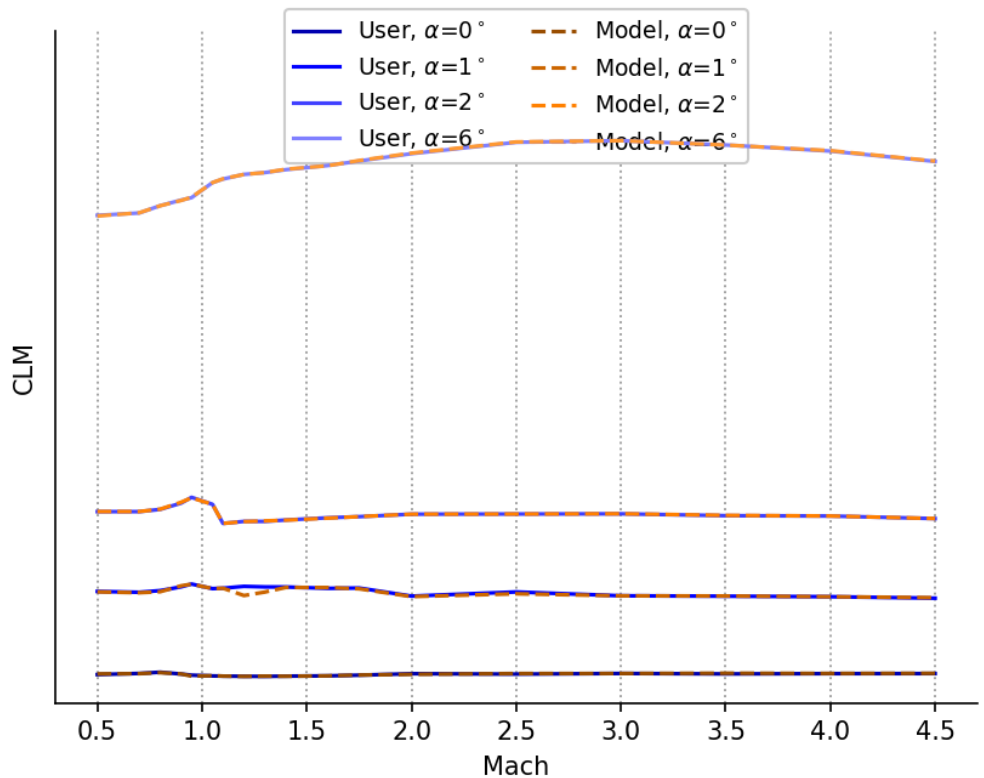
CORE

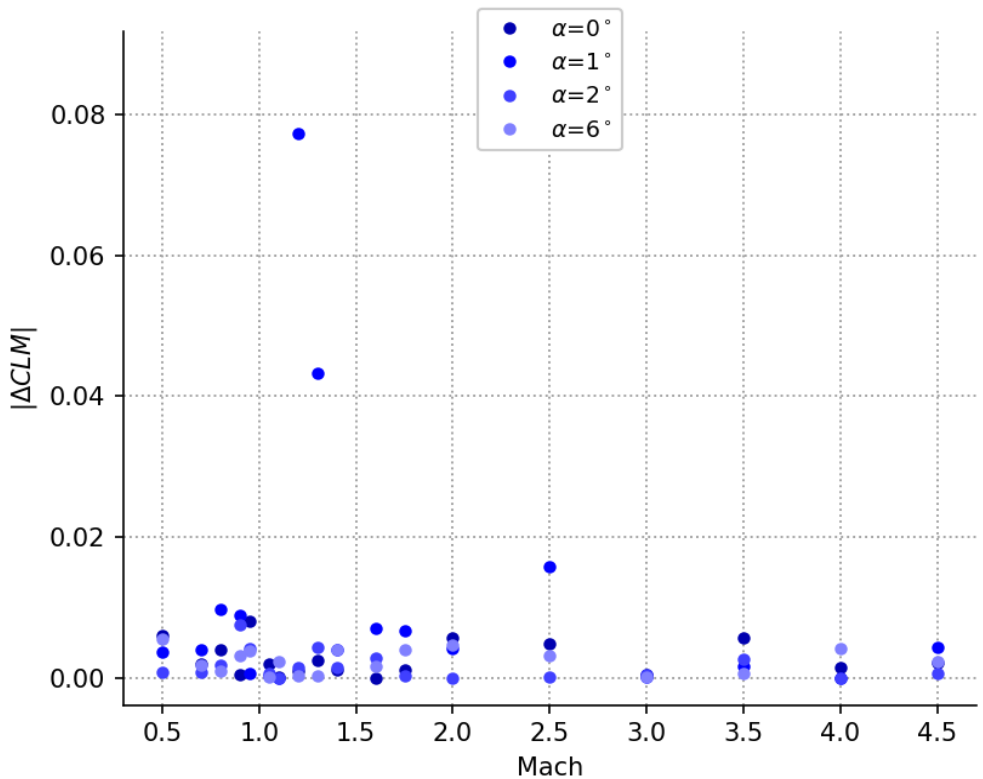




Results: User vs Model

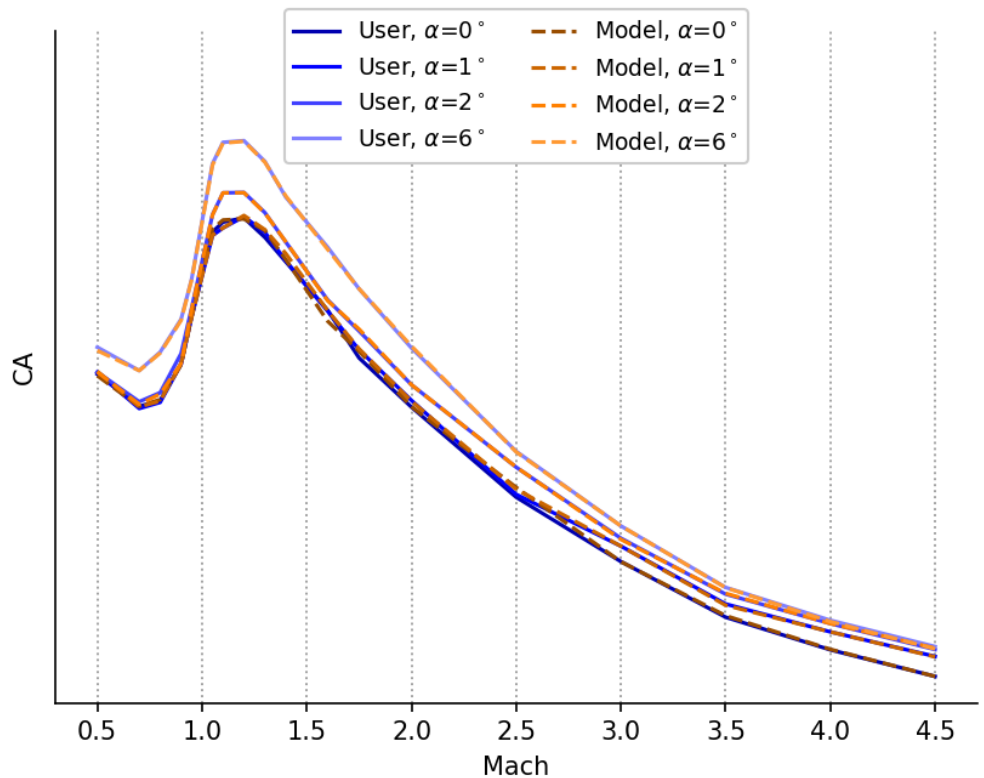
CORE

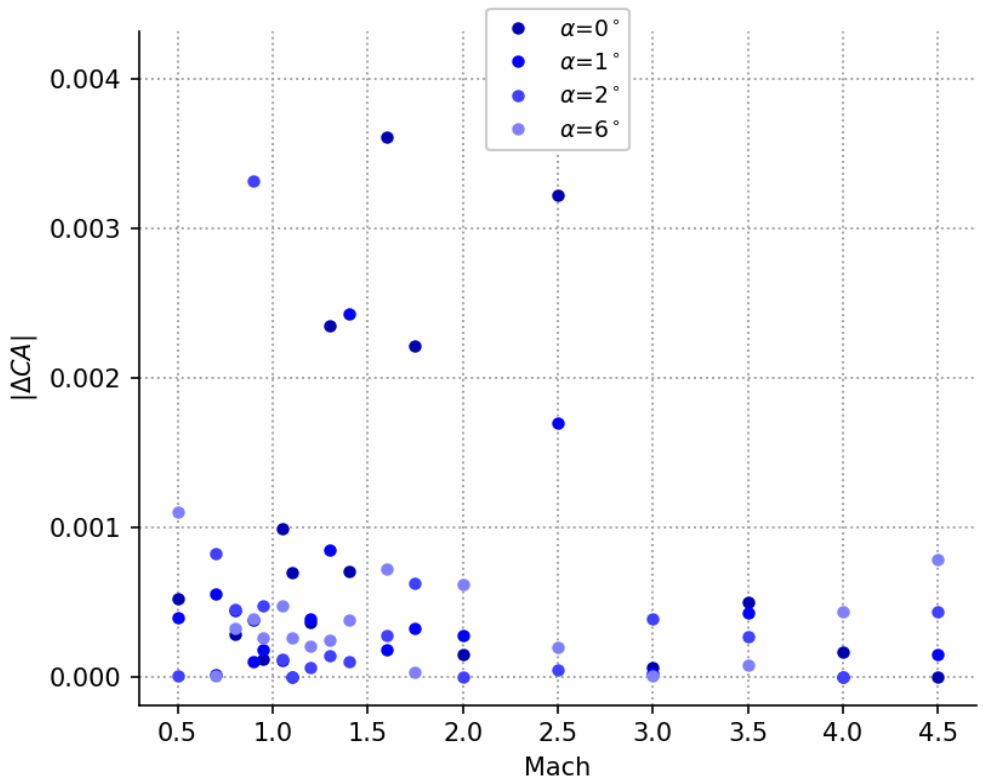




Results: User vs Model

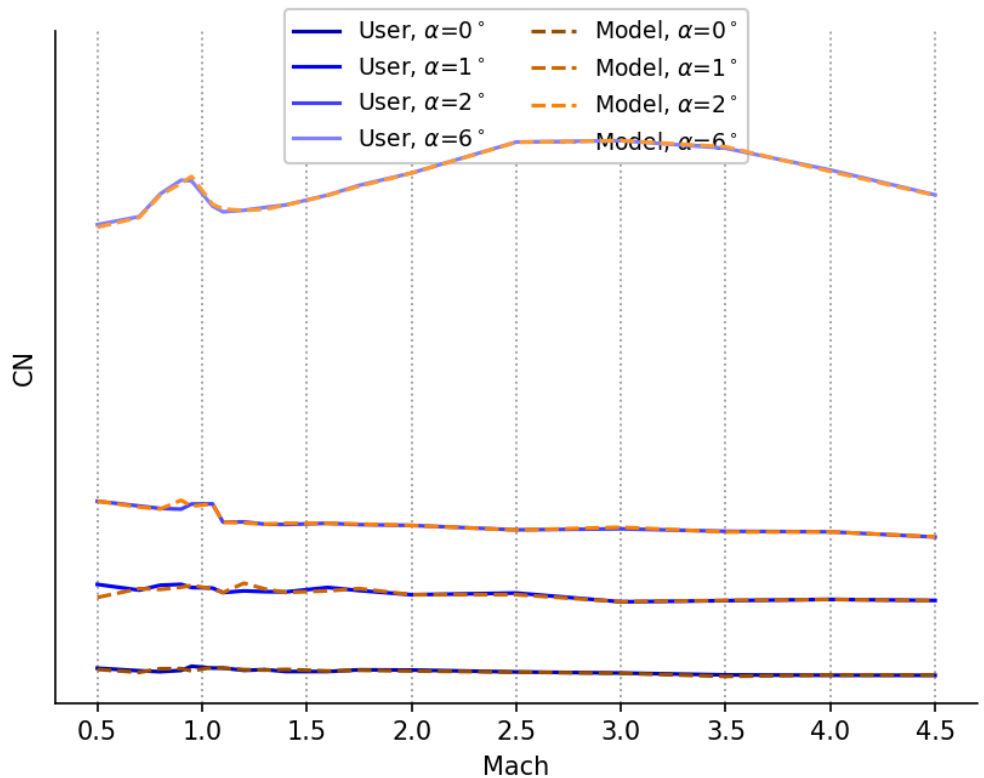
RSRB

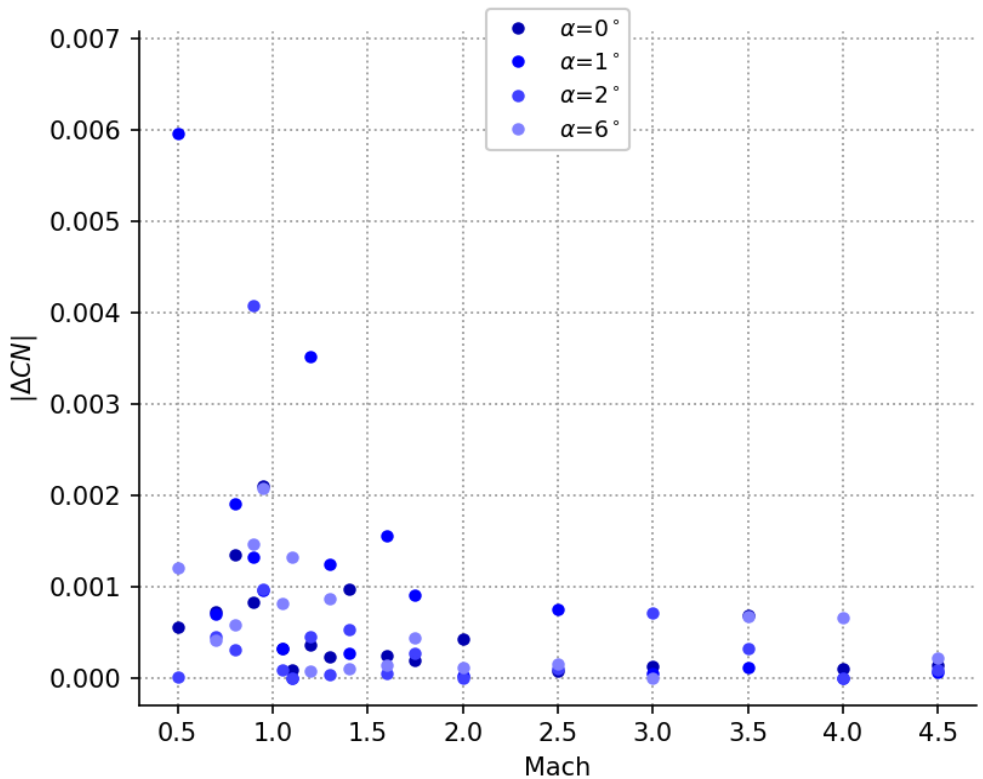




Results: User vs Model

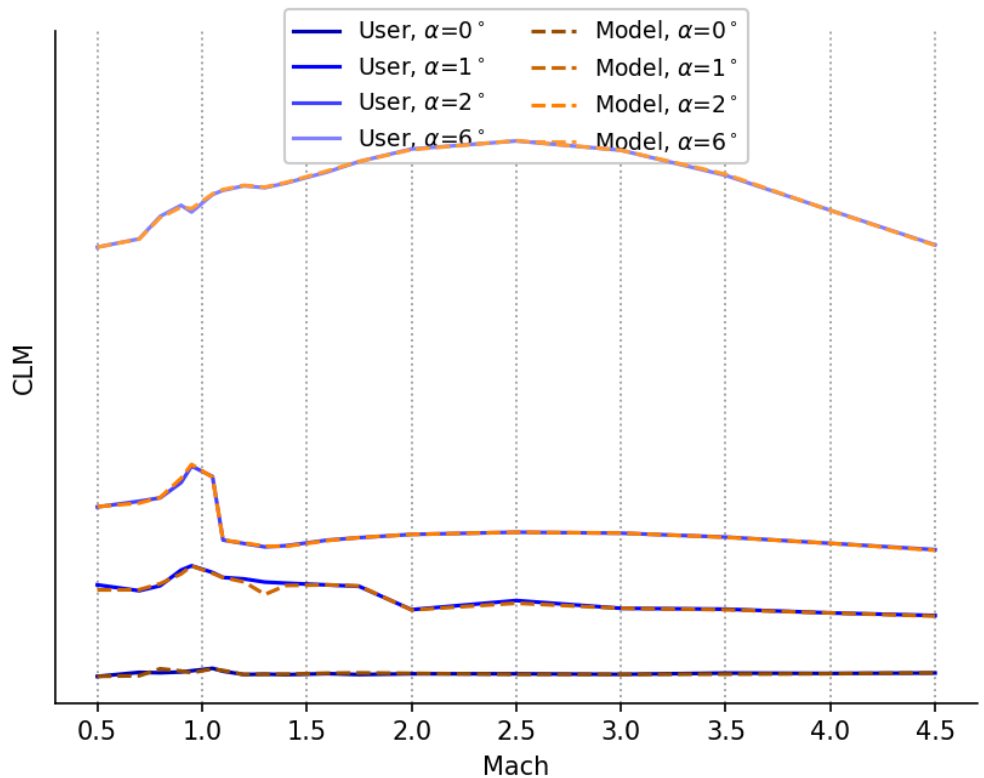
RSRB

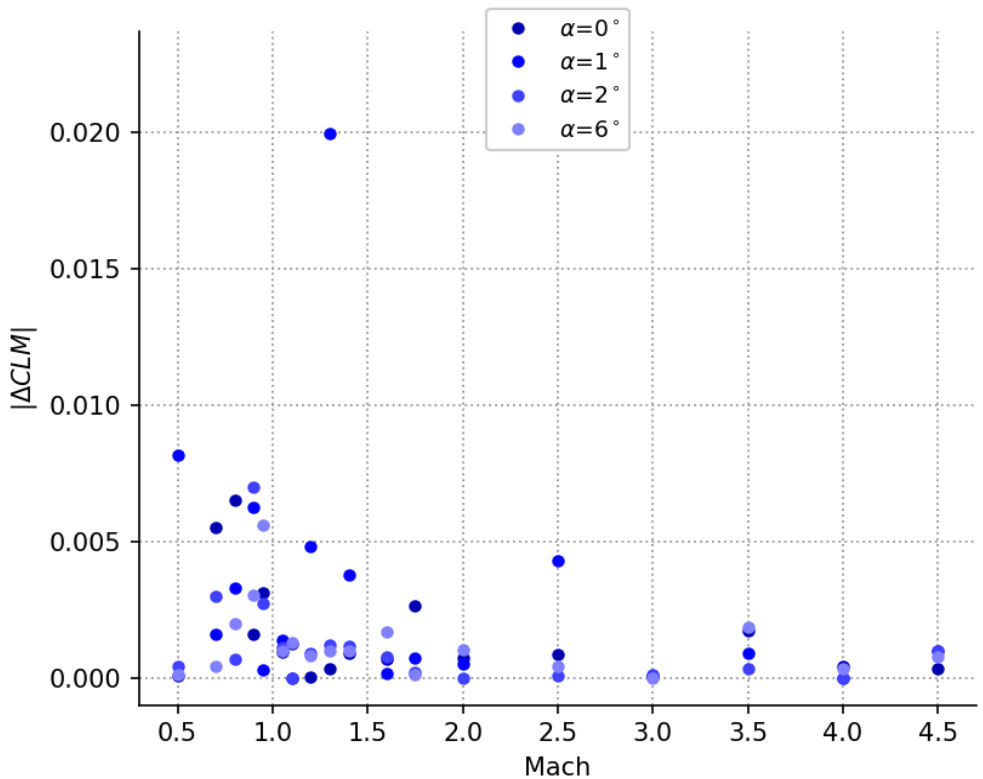




Results: User vs Model

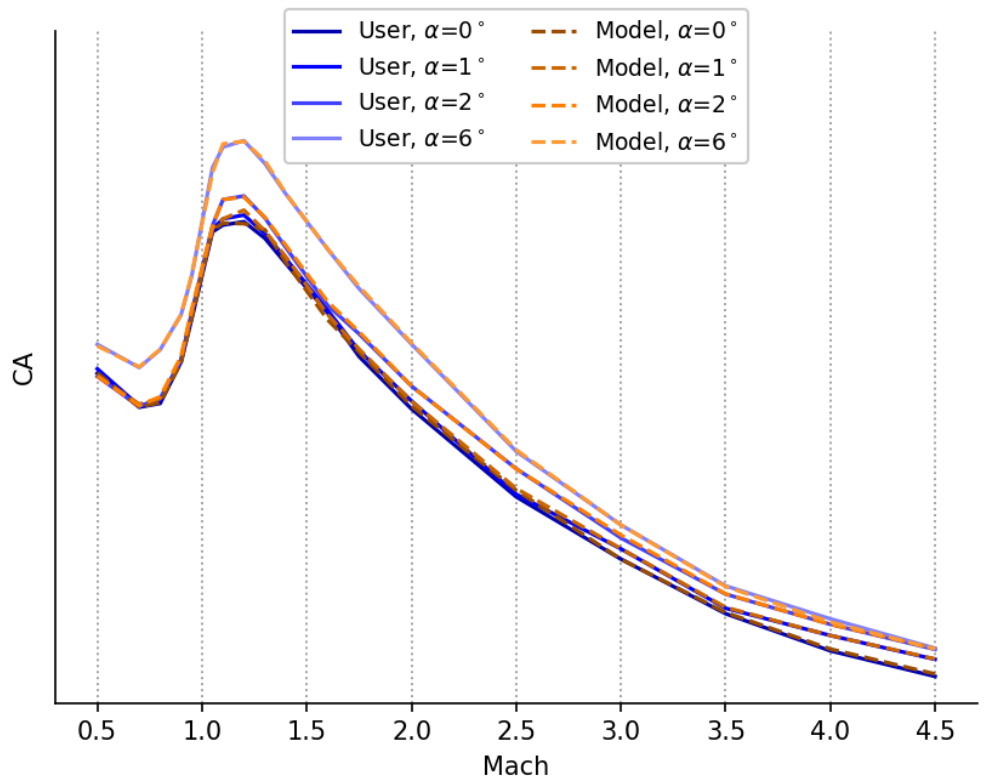
RSRB

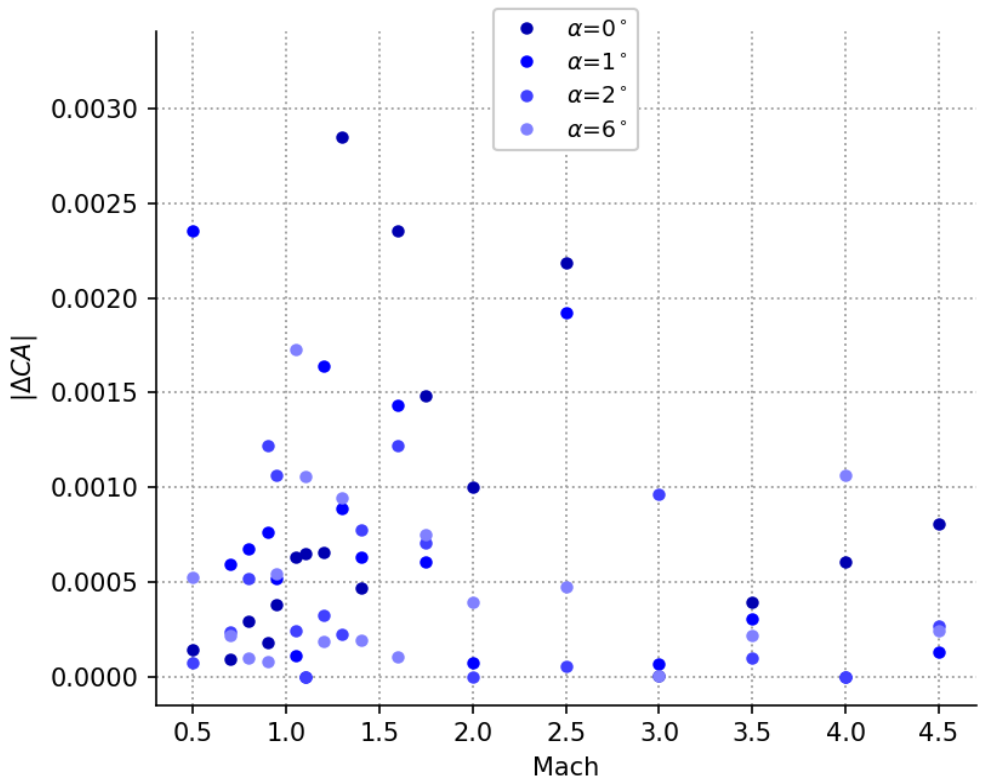




Results: User vs Model

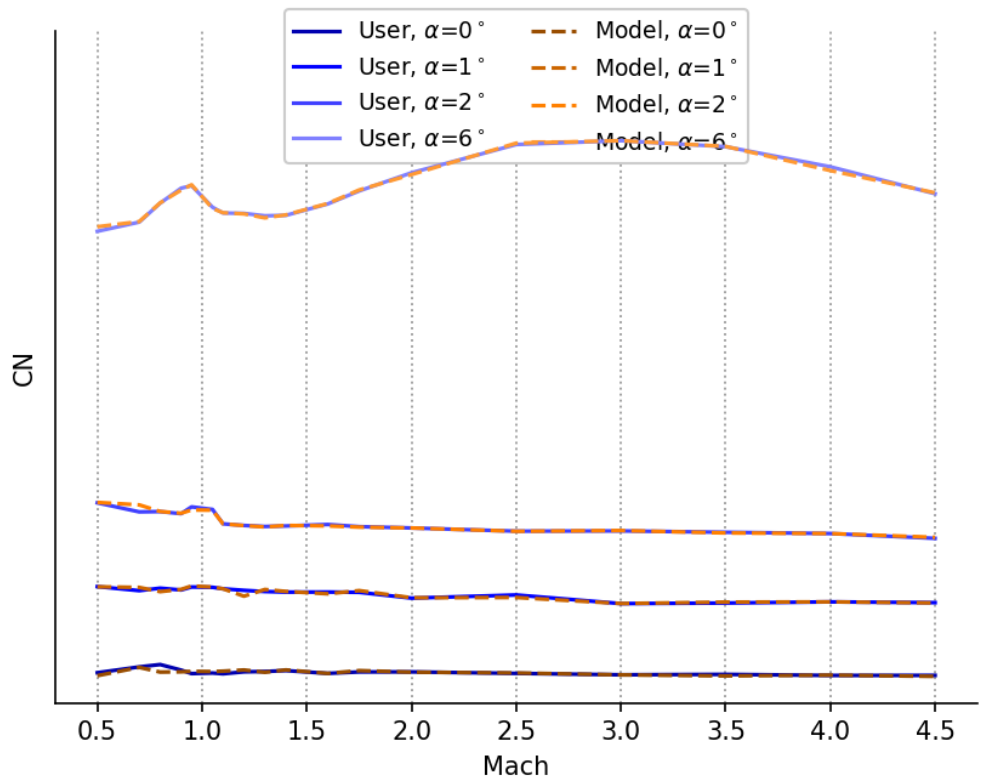
LSRB

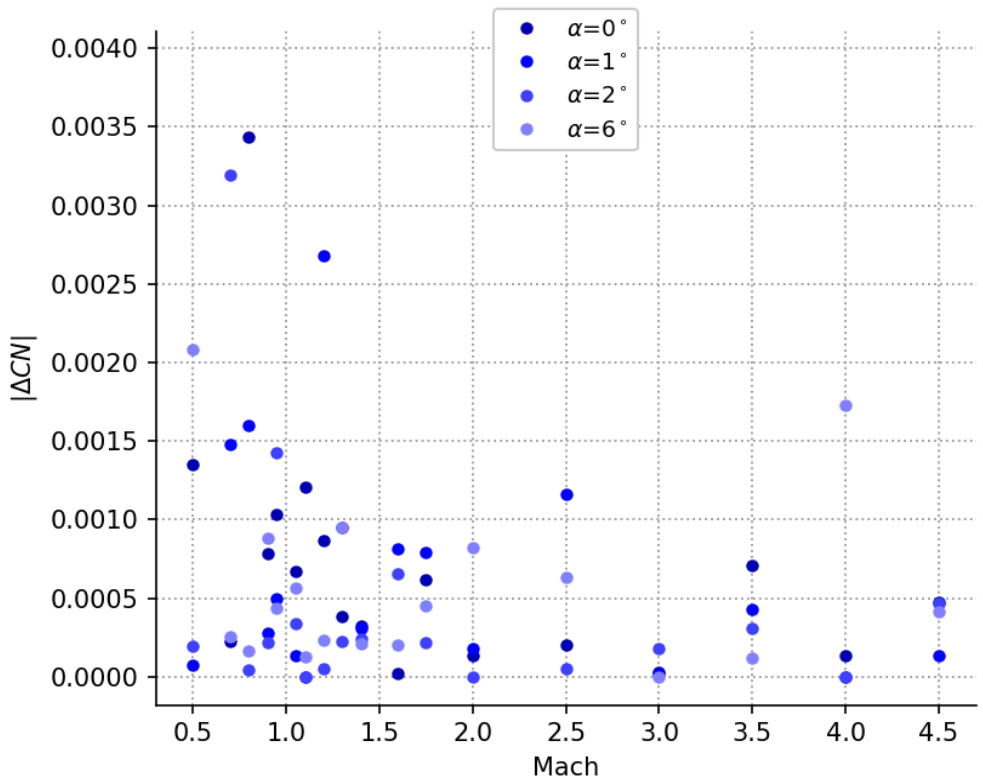




Results: User vs Model

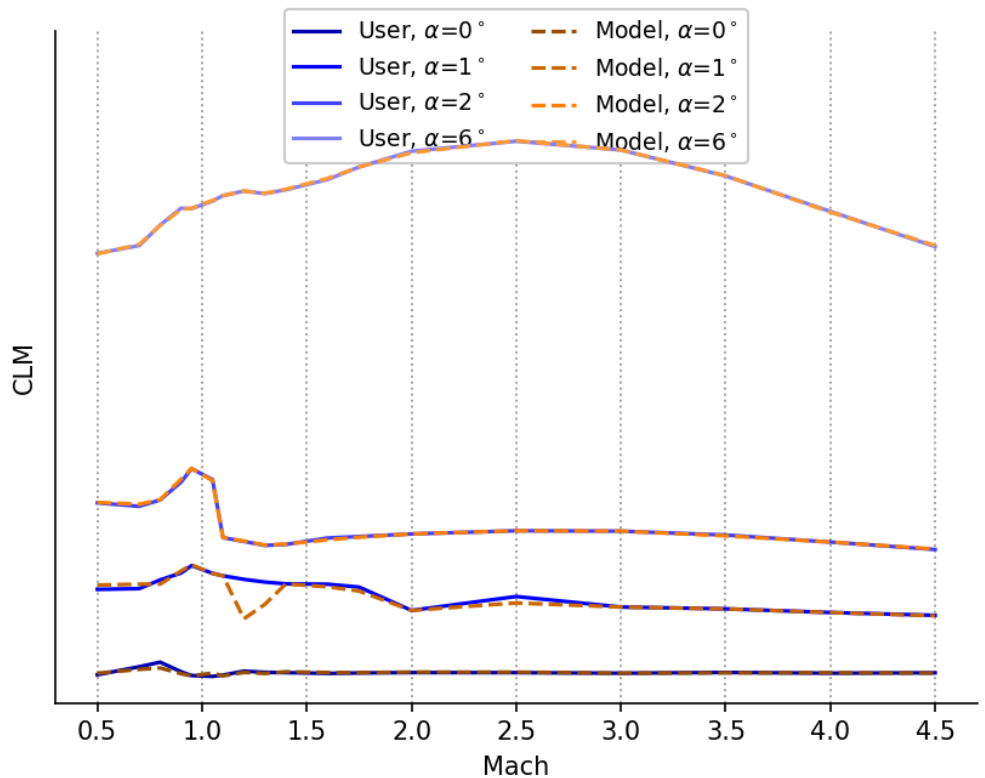
LSRB

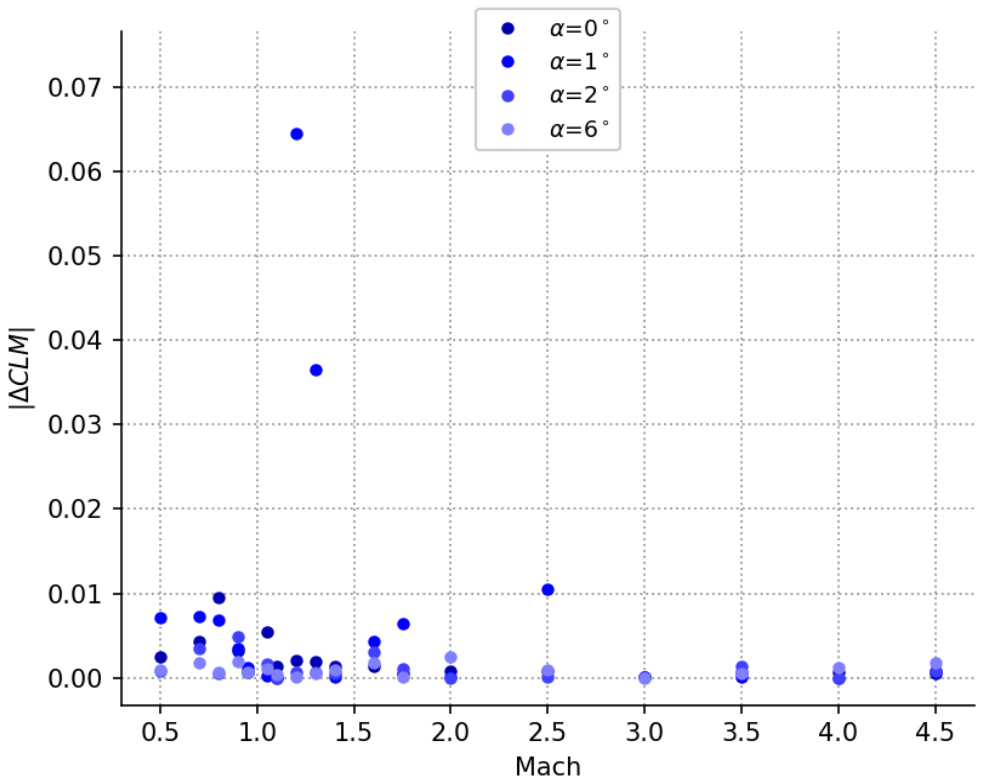




Results: User vs Model

LSRB





Summary

Machine learning techniques were shown to **reliably classify** complex iterative histories and be **more capable** than traditional convergence criteria alone.

Future Work

1. More generalized model
    - Probability sampling of iterative history
    - Feature variable selection study
2. Additional classes
    - Switch mesh settings
    - Switch solver settings
3. Additional Solvers
    - Cart3D
    - OVERFLOW

Acknowledgements

NASA High-End Computing

- NASA Advanced Supercomputing Division
- High-End Computing Capability (HECC)
- Pleiades, Electra, and Aitken supercomputers

Ames SLS CFD Team

- Stuart Rogers, Derek Dalle, Jamie Meeroff, Guy Schauerhamer, Aaron Burkhead, Josh Diaz

CFD software highlighted in this work

*FUN3D (Langley)